Barriers That Blind Programmers Face

Textual Navigation

• Programmers and students usually use visual aids called Integrated Development Environments (IDE) to help write programs, but blind programmers might not get as much information as their sighted counterparts. (StructJumper)
• Context switching between lines of code or debugging might be inefficient for blind students. (StructJumper)
• Screen reader cannot help generate a general picture of the code structure for blind programmers. (StructJumper)
• Complicated textual code will eventually converted to a hierarchical structure to make navigation and debugging easier.

Spatial/Graphical Navigation

• Screen reader cannot "read" graphs. (Blocks4All)
• Drag-and-drop programming is highly visual; therefore unavailable to blind learners. (Blocks4All)
• Some important graphs, often as node-link diagrams, are widely used in Computer Science, but they are often inaccessible to students and professionals. (GSK)

Existing Solutions

Textual programming needs spatial navigation, too!

Textual Navigation

• StructJumper:
  • A plugin to existing programming tool that creates hierarchical tree to help blind programmers navigate the code.
  • It is compatible with one of the popular programming visual aids called Eclipse.
  • It makes both navigating the code and understanding where they are within the code easier for blind programmers.
  • Complicated textual code will eventually converted to a hierarchical structure to make navigation and debugging easier.

Spatial/Graphical Navigation

• Blocks4All:
  • A drag-and-drop programming language primarily designed for children.
  • It simplifies the syntax of programming languages.
  • Universal designs means it is accessible for both sighted and blind children.
  • It is based on touch screen, which can be explored using speech and sound compatible to screen reader.

Graph Sketching (GSK):

• It has two views – Connection View and Grid View
• The connection view uses graph – a data structure.
• Its connection view is accessible for both sighted and blind programmers.
• The grid view adopts chessboard idea.
• Both connection and grid view can guide blind people to add/remove/edit a node or a connection.
• Its properties windows can help turn the graph into text to make it readable for screen reader.

Quorum Programming Language:

• A programming language designed to help people with different disabilities.
• It has tutorial for novice programmers.
• It uses screen reader in order to create an audio programming language environment.
• The developers of this language simplified the syntax to make it more accessible for blind students.

What’s Next?

• Solutions above have some shortcomings when it comes to circuits design:
  • Hierarchical trees usually have limited amount of layers; therefore when dealing with large chunks of code, StructJumper might be inefficient.
  • Block4All might not be helpful when it comes to complex circuit design, since it is designed for children.
  • GSK treats nodes and connections equally, but in circuit design, wires and endpoints/nodes have different functions.
• We plan to focus on improving the accessibility of spatial/graphical navigation, because:
  • Programmers tend to come up with schematics to help them structure their textual code.
  • In circuit design, textual programming like Verilog also has outputs in graphical form.
• Graphic programming tends to be easier for beginners to understand.

We choose Logisim, a digital circuits design language, as our primary language to modify.
• We must make Logisim compatible with screen readers.
• We plan to adopt some of the ideas of GSK, specifically the chessboard idea from the grid view.
• We must modify GSK to distinguish different types of connections and nodes.
• In logic circuits we need a different type of node for each component (e.g., AND, OR, NOT).
• Since each port of a component has a specific purpose, a node’s connections are not interchangeable and so connections must be named or ordered.

References

• Quorum: https://quorumlanguage.com/

Acknowledgement

• This research is supported by ICRU.
• Special thanks to Professors Kyle Rector and Brandon Myers for the support, advice, and help.
• Special thanks to Dr. Catherine Baker for the helpful suggestions on the research.